

# MINIX+1.0 操作系统实验 环境设计与实现\*

黄祥喜

(中山大学计算机科学系, 广州 510275)

**摘 要** MINIX+1.0 操作系统实验环境是我们对美国知名学者 A. S. TANENBAUM 1986 年编制的分时多任务多用户操作系统 MINIX 研制的改进版。和 MINIX 初版相比, MINIX+1.0 具有进程对换, 假脱机 I/O, 扩展内存的管理等许多新功能。本文从功能结构, 实现原理, 用户环境等几个方面对 MINIX+1.0 作了论述。

**关键词** 操作系统, 实验教学, 计算机教育, MINIX, MINIX+1.0

**分类号** TP316

操作系统作为计算机专业课程, 在高校教育中一直倍受重视。由于没有一个好的实验环境, 操作系统实验一直是操作系统教学中的薄弱环节。

目前, 操作系统的实验方式, 可归纳为以下 3 种: ①模拟环境下的实验; ②实用系统的分析; ③基于模型系统的实验。

综合比较上述 3 种实验方式, 不难得出结论: 使用模型系统应是最优的。它专为操作系统的教学而设计, 巧尽心思地将有关的理论部分实现出来, 包含了第一种方式: 模拟环境下的实验的各种效果。同时, 模型系统又具有第二种方式: 实用系统分析的部分功能, 如用户界面和系统调用可实现得与某个实用系统一样。

从 1989 年开始, 我们相继研究了 SOS<sup>[1]</sup>, XINU<sup>[2]</sup>, MINIX<sup>[3]</sup> 等系统。1990 年确定将 MINIX 作为我们的实验环境的基础, 对它进行修改和扩充。经过近两年的工作, 终于研制了 MINIX+1.0。1992 年 4 月, MINIX+1.0 操作系统实验环境通过国家教委委托中山大学组织的专家鉴定。鉴定结论: 本系统具有国内领先水平, 并在功能和结构上优于 MINIX 初版, 本文对 MINIX+1.0 操作系统实验环境的实现原理作了全面论述。

## 1 MINIX+1.0 操作系统实验环境的功能特点

MINIX+1.0 建立在 IBM-PC 系列微机之上, 其用户对象是正在学习操作系统课程

收稿日期: 1992-12-22

\* 广东省科学基金资助项目

的学生. 为了较完整地体现操作系统中的基本观点与概念, 同时又考虑到是在微机上实现, MINIX+1.0 具有以下一些功能特点:

- (1) 在进程管理和调度上, 它采用多重队列管理和轮回调度相结合的原则.
- (2) 在进程通信中, 采用消息机制.
- (3) 在设备管理中, 对一类设备编制相应的驱动程序, 同时规定统一的访问方法(用消息机制), 使之实现“与设备无关性”.
- (4) 在内存管理方面, 采用多道程序的链接表管理方式, 同时支持进程对换, 实现主存扩充.
- (5) 在文件系统方面, 实现同 UNIX 相一致的树形目录管理, 同时提供文件保护机制.
- (6) 在用户界面方面, 实现同 UNIX V7 相一致的外壳 shell, 提供管道、重定向等机制以及一套命令和工具. 另外, 还实现了功能优于 UNIX V7 的多设备假脱机输入输出.
- (7) 在体系结构方面, 将“层次结构”与“顾客-服务员”模式两者相结合. 各模块之间采用消息机制通信.

## 2 MINIX+1.0 操作系统实验环境的构成

完整的 MINIX+1.0 系统分为二种版本:

- (1) 286 以上版: 由 4 张 1.2MB (MS-DOS 格式) 软盘组成. 其中包括启动和根系统盘 (1 张), 用户盘 (1 张)、源程序盘 (2 张).
- (2) 8088/8086 版: 由 11 张 360KB (MS-DOS 格式) 软盘组成. 其中包括启动盘 (1 张), 根系统盘 (1 张), 用户盘 (3 张), 源程序盘 (6 张).

MINIX+1.0 源代码由 C 语言及汇编语言组成. 其中 C 源代码共 14329 行, 汇编源代码 789 行. MINIX+1.0 的引导盘由取名 build 的实用程序建立, 它以一种特殊的方式把 6 道程序连在一起. 这 6 道程序是: ①引导块 (bootblok), ②内核 (kernel), ③存储管理 (mm), ④文件系统 (fs), ⑤系统初启程序 (init), ⑥文件检验器 (fsck).

## 3 MINIX+1.0 操作系统实验环境的体系结构

MINIX+1.0 的构造分为 4 层, 每一层执行意义明确的职能 (参见图 1).

最底层捕捉一切中断和陷阱, 并且为较高的那些层提供一个利用消息进行通信的独立顺序进程模型.

第二层包含 I/O 任务, 配置了各种设备驱动程序. 各任务的职能如下:

RAM	TASK: RAM 盘设备驱动程序
FLOPPY	TASK: 软盘设备驱动程序
WINICHE	TASK: 温盘设备驱动程序
CLOCK	TASK: 时钟设备驱动程序

TTY TASK: 终端设备驱动程序

PRINTER TASK: 打印机设备驱动程序

SWAP TASK: 对换系统驱动程序

SYSTEM TASK: 利用标准的消息机制使内核与文件系统和存储管理程序通信

这一层的所有代码与第一层的代码连结在一起成为单独的一个程序,称为内核。

层号

(4)	初启进程	用户进程	用户进程	用户进程	...	用户进程
(3)	存储管理		文件系统			服务进程
(2)	盘任务	对换任务	时钟任务	系统任务	...	1/0 任务
(1)	进 程 管 理					

图 1 MINIX+1.0 的体系结构

Fig. 1 Architecture of MINIX+1.0

第三层包含两个进程,向用户提供有益的服务.存储管理 MM 实现涉及存储管理的全部 MINIX+1.0 系统调用,比如 FORK, EXEC 和 BRK;文件系统 FS 实现所有的文件系统调用,比如 READ, MOUNT 和 CHDIR.

最后,第四层容纳所有的用户进程——SHELL 外壳、编辑程序、编译程序、假脱机以及用户编写的各种程序。

## 4 MINIX+1.0 操作系统实验环境的实现原理

### 4.1 MINIX+1.0 中的进程及进程管理

在多任务操作系统中,进程是一个重要的概念. MINIX+1.0 操作系统使用了这个概念,并对进程的实现在进行了改进。

MINIX+1.0 本身是一组进程,其内核分割为若干个模块.采用消息传递作为进程间通信的唯一原语(包括 MINIX+1.0 内部各个进程之间的通信,以及 MINIX+1.0 内部各个进程跟用户进程的通信)。

4.1.1 MINIX+1.0 系统中的进程表 在 MINIX+1.0 系统中,进程管理、存储管理及文件管理各自由系统里的相应模块处理.因此,MINIX+1.0 中的进程表是分区的,每个模块维护自己要用的域.目前,MINIX+1.0 的进程表由 3 个相对独立的部分构成,它们是:① kernel 中的进程表;② MM 中的进程表;③ FS 中的进程表。

4.1.2 MINIX+1.0 系统中的进程状态 在 MINIX+1.0 系统中,进程可能处于以下 5 种状态:① 运行;② 在内存中就绪;③ 在内存中睡眠;④ 换出且就绪;⑤ 换出且睡眠.图 2 是 MINIX+1.0 系统的进程状态转换图。

4.1.3 MINIX+1.0 系统中的进程通信 MINIX+1.0 系统中的进程(包括任务在内)是运用会合原理,通过定长消息的交换实现的.当一个进程做一个 SEND 时,内核查看

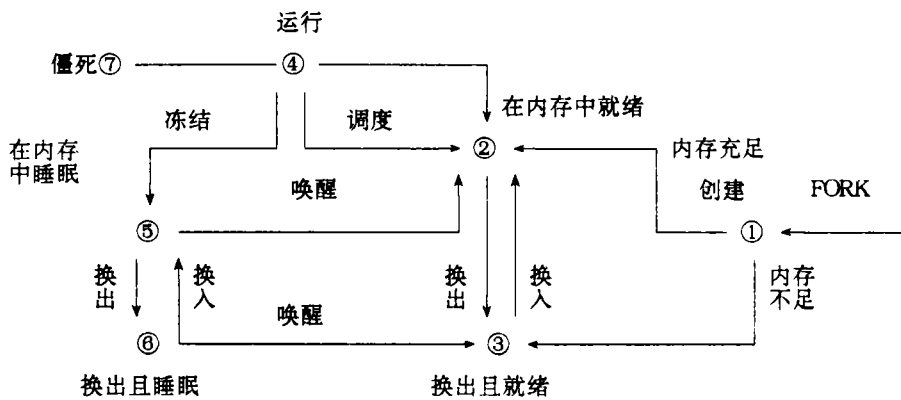


图 2 对换系统下的进程状态转换图

Fig. 2 Transition diagram of process states in swapping system of MINIX+1.0

目标是否在等待来自该发送者(或任一发送者)的消息。若是,这条消息则从发送者的缓冲区抄到接收者的缓冲区里,两个进程都标记为可运行的;否则,该发送者被标记为冻结,并被排入等候向这位接收者发送的进程队列中。

由于 MINIX+1.0 具有进程对换管理功能,系统中进程的映像可能被换出到对换设备上,这迫使 MINIX+1.0 引进了消息缓冲机制,为每个进程开辟一个信箱,每个信箱只能容纳一个信件(一个消息),这样,它们是在内核中的一个消息数组。当一个消息被发送到处于“换出”状态的该消息的接收者时,内核将会把该消息拷贝到接收者的信箱中(而不是拷贝到接收者进程映象中的消息结构里)。并把两个进程都标记为可运行的(而不是冻结发送者)。尔后,当对换子系统将该接收者换入内存时(这是必然的,因为该接收者总会从“换出且睡眠”状态转换成“换出且就绪”状态),对换子系统将会发现该进程的信箱中有信件(即消息)。于是,对换子系统将它拷贝到接收者进程映象中的消息结构里。这样,使消息的传递与目的进程是否换出无关。

在新的消息机制中,每个进程的邮箱只能容纳一个信件(消息)。这使新的消息机制仍然体现着会合原理(当一被换入的进程在取走它的信件之前不能接收消息)。

4.1.4 MINIX+1.0 的进程调度 MINIX+1.0 的进程调度采用三级队列方式。第一级为 I/O 任务;第二级为服务员进程;存储管理及文件系统;第三级为用户进程。在每一级的内部,采取轮回算法,任务享有最高优先权,存储管理及文件系统其次,用户进程居末。

在挑选要运行的进程时,调度程序按一、二、三的顺序检查 3 个队列中是否有就绪进程,若有则选择一个激活之。在每一次时钟中断时会检查当前的进程是否是一个已运行一个时间片(100ms)以上的用户进程。若是,则调度程序会检查是否有另一个进程在等待 CPU;若有则把当前的进程退到调度队列的末尾去而运行排在队首的进程。而 I/O 任务、存储管理及文件系统进程则决不会被其它进程逼退。直到它们自己结束。

4.1.5 MINIX+1.0 系统中的信号处理 在 MINIX+1.0 中,信号有两种产生途径,用户进程进行 KILL 系统调用或由内核产生。但不管什么信号源,存储管理都用同样的

方法来处理所有的信号. 对于发向每一个进程的信号, MM 都要进行各种合法性检查. 一旦条件成熟, 信号就可以发送. 如果信号被捕捉, 一消息将由 MM 发送到内核的系统任务. 当系统任务接收到此消息时, 它将在捕获该信号的进程栈中存放程序状态字, CS 寄存器, 程序计数器和信号号码等 4 个字并把控制权转交给该进程.

具有进程对换管理功能的 MINIX+1.0 系统并不全是即时响应进程信号的. 当一个换出进程捕捉信号时, 系统将保留该信号, 待到该进程换入时再发送. 具体地, MM 在进行信号处理时, 它将检查信号的目标进程的状态. 如果该进程正处于“换出”状态, 则 MM 并不通知内核, 而是将该信号的号码保存在目标进程的进程表里. 在一个进程被换进内存时, MM 将从该进程的进程表中检查该进程换出期间是否曾捕捉过信号. 若是, 则发消息通知内核进行信号处理.

#### 4.2 存储管理

MINIX+1.0 在存储管理技术上较 MINIX 先进, 它支持二级存储管理. 与 UNIX 不同的是, MINIX+1.0 的存储管理并不是内核的一部分, 而是作为存储管理服务进程在用户空间运行. 把存储管理移出内核是策略和机制的分离, 进程的控制由存储管理完成(策略), 进程的调度由内核完成(机制). 这种分离使得修改存储无需牵扯操作系统的低层部分, 且更加容易、可靠.

MINIX+1.0 的存储管理功能是由存储管理进程 MM 实现的. MM 不单完成了二级存储管理, 还实现了有关存储管理的系统调用. MM 对二级存储器的管理机制是相类似的. 它为二级存储器维护一张自由空间表. 在进行存储分配时, MM 使用最先适配算法, 寻找自由空间表, 找到并分配第一个满足要求的自由块.

MINIX+1.0 引进了进程对换系统, 图 3 是对换系统的体系结构图.

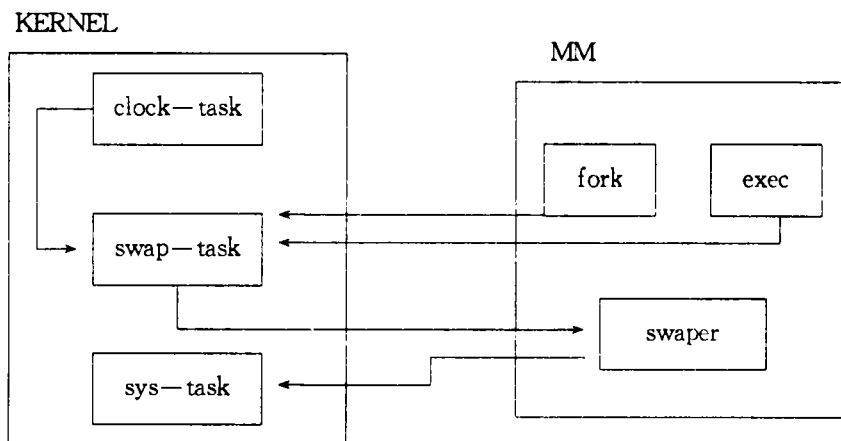


图 3 MINIX+1.0 对换系统的体系结构图

Fig. 3 Architecture of swapping system of MINIX+1.0

在 MINIX+1.0 中, 操作系统分成 3 个部件: kernel, MM, FS. 而用于实现进程模型

的进程表也分布于三者之间. 这决定了与进程控制息息相关的进程对换系统不可以象 UNIX 一样集中实现于其中一个部件中. 相应于 UNIX 的对换进程, MINIX+1.0 的内核含有一个对换任务 swap-task. 该任务定时地被时钟任务唤醒, 并根据当前各用户进程的状态, 依照一定的对换算法, 决定哪些进程应该被换入, 哪些进程可能被换出, 并据此建立换入进程队列和换出进程队列. 具体地, 将当前状态为“在内存中睡眠”的进程按睡眠时间递减顺序加入换入进程队列, 将当前状态为“换出且就绪”的进程按就绪时间递减顺序加入换出进程队列. 然后, 通知 MM 开始进程对换动作. 至此, 对换任务已完成其主要工作. 而进程对换的实质: 在内存与对换区之间传送进程映象这一关键动作由 MM 完成. MM 提供了 swap-task 使用的系统调用 DO-SWAP. 该系统调用根据一定的进程对换算法, 进行进程对换的工作: 试图将换入进程队列上的队列逐一换入, 必要时, 从换出进程队列中选其一换出, 以腾出内存空间. 直至出现以下两种情况: ① 换出进程队列已空; ② 找到应换入的进程, 但无足够内存空间, 且换出进程队列空或换出动作失败. 如此反复, 对换系统在后台进行二级存储调度, 但就用户而言, 这一过程完全是透明的.

#### 4.3 I/O 设备管理

MINIX+1.0 的 I/O 系统是一个分层结构, 参见图 4. 通过严格分层, 使得设备无关性在 I/O 软件中得到很好地体现. 由中断处理程序负责直接的硬件操作. 由设备驱动

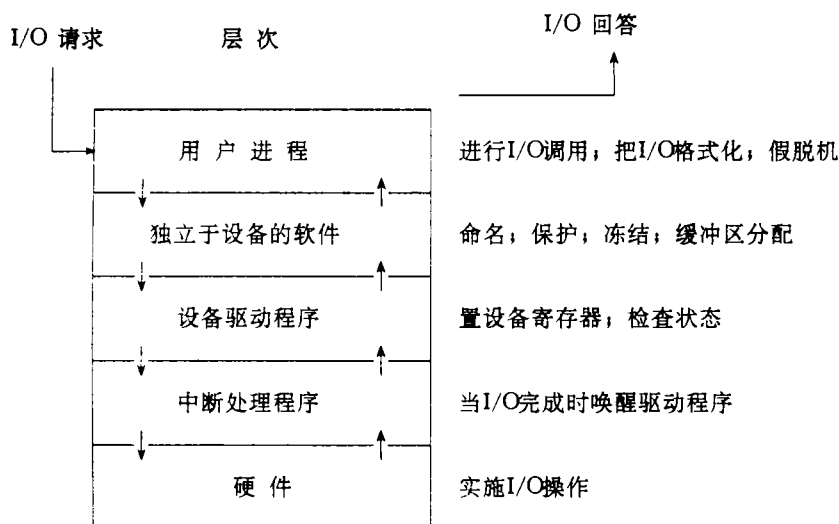


图 4 MINIX+1.0 I/O 软件的结构

Fig. 4 Structure of I/O software of MINIX+1.0

程序启动 I/O 设备. 它是作为一系列进程而嵌入系统核心的. 文件系统担当着独立于设备的软件的角色. 它负责设备的命名、保护、缓冲以及分配的工作. 在这一层, 所有的块设备如硬盘、软盘等都能安装在文件系统层次中的任意位置, 一切文件和设备在这里都用相同的工具——路径名来检索. 用户进程空间的 I/O 软件包括实现系统调用的库过程

及假脱机进程等。

MINIX+1.0 的一个重要特点是,所有的设备驱动程序均以进程的形式嵌于核心,这与 UNIX 的单片系统不同。这样的好处是明显的:即高度的模块化,易于修改和扩充。但由于是通过消息传递来进行进程间通信,故效率比 UNIX 的单片系统差。

#### 4.4 文件系统

文件系统是文件和目录加上施于它们的操作的集合。它是操作系统与用户关系最密切的部分。对于大多数用户来讲,能否方便地使用一个操作系统以及它的可信赖程度如何,就在于它的文件系统与用户的界面是否良好,以及其可靠性是否得以保证。

MINIX+1.0 的文件系统的内部结构与 UNIX 兼容,它具有以下的特点:

(1) 文件系统的组织为分级树型结构。MINIX+1.0 的文件系统基本上是一棵倒向的树,树的根是根目录,叶结点是文件,在根和叶之间的结点是目录。

(2) 可以动态地装卸文件卷(硬盘或软盘)。对于位于硬盘或软盘上的文件系统,可以动态地安装到原有的文件系统上,形成一个更大的分级结构文件系统,当然也可以将它卸下来,以安装另一个文件系统,系统显得方便、灵活。

(3) 设备无关性。不论文件、目录表还是外部设备,在用户进程眼中均有统一的语义和语法,置于同样的保护机制下,用户可对它们进行同样的操作,亦即统一命名。

(4) 采用 i 结点记录文件的说明部分。与 UNIX 系统一样,MINIX+1.0 采用了把文件名和文件说明部分分开的方法。它们分别作为目录文件和 i 结点表中的一个表项,这不仅可加速对文件的检索过程,减轻通道的 I/O 压力,而且还可给文件的联结和共享带来极大方便。

(5) 采用直接—间接的文件映象法。文件长度不超过 7 个盘块时,采用了直接映象方法,在相应 i 结点的指针中直接存放文件所在的盘块号;而对于长度超过 7 个盘块的文件,则采用了间接映象方法,即在相应 i 结点的盘块指针中存放的是存放文件盘块号的盘块号。MINIX+1.0 中最多有二重间接指针,使得文件长度最大可达到 262 兆。

(6) 采用位图来管理空闲 i 结点及空闲盘块。这是 MINIX+1.0 系统不同于 UNIX 系统的地方。它用 0, 1 来表示每一 i 结点或逻辑块的使用与否,其管理算法在速度上是最快的。

在 MINIX+1.0 系统中,当用户要读/写文件(包括各种设备)时,就发送一消息给文件系统,文件系统处理用户请求并把结果返回给用户进程。这实质是文件系统完成一项系统调用的过程。

MINIX+1.0 的文件系统不象 UNIX 那样作为内核的一部分,而是作为一个进程在用户空间执行。它与存储管理进程享有同等的优先级。它的源程序可以根据需要进行修改及扩充,与 MINIX+1.0 的其它部分分开调试。作为一个完善的系统,它还很容易移植到其它含有 C 编译器的计算机上。

## 5 MINIX+1.0 系统的用户环境

MINIX+1.0 操作系统实现了与 UNIX V7 兼容的用户环境,具体有以下几个方面:

(1) 系统调用. MINIX+1.0 实现了 UNIX 中的大部分调用, 共提供了 44 个系统调用. 其调用方法与功能均与 UNIX 相一致. 同时, MINIX+1.0 还提供了 7 个系统内部管理用的系统调用.

(2) shell. MINIX+1.0 的 shell 与 UNIX 一致, 用户通过命令的方式同操作系统打交道; 同时, shell 还提供了重定向、管道等机制以及判断、循环等控制结构, 使用户能以文件和 shell 变量为基础, 编制被誉为“超高级语言”的 shell 程序.

(3) 公用命令和工具. MINIX+1.0 实现了 UNIX 大部分的 shell 命令, 共 72 个. 同时, MINIX+1.0 还提供了一组有自己特色的工具.

(4) 假脱机系统. MINIX+1.0 提供了一个功能齐全假脱机系统, 可以对任意已定义的慢速设备进行假脱机输入输出, 并且可查看用户提交的任务状态、假脱机设备状态, 以及取消某一假脱机任务等.

## 6 MINIX+1.0 同 MINIX 的比较

MINIX+1.0 是在保持 MINIX 原有特色的基础上进行改进和扩充而形成的, 同时, MINIX+1.0 也修改了 MINIX 中若干失误之处, 使之更加完善.

对于 MINIX+1.0 同 MINIX 的比较, 我们重点放在扩充和修改上:

(1) 由于 MINIX+1.0 支持进程对换, 实现了虚拟存储, 因而对于 MINIX 来说, 主存有限, 而 MINIX+1.0 却突破了主存的限制, 同时可运行的进程数从 16 个(MINIX) 扩充到 50 个, 甚至可更大.

(2) MINIX+1.0 具有假脱机系统, 可支持假脱机输入输出, 同时对假脱机 I/O 设备进行的管理; 而 MINIX 没有此功能.

(3) MINIX+1.0 支持 AT 机扩展内存(extended memory); MINIX 没有支持.

(4) MINIX+1.0 支持 386, 486 机器; MINIX 没有支持.

(5) MINIX+1.0 支持 1.44M (MS-DOS 格式) 3.5 英寸盘; MINIX 不支持.

(6) MINIX+1.0 支持引导盘和根系统盘的重置; 而 MINIX 引导盘和根系统盘只能固定在驱动器 0 上.

(7) MINIX+1.0 扩充和修改了 MINIX 的库函数, SHELL 和命令.

总之, MINIX+1.0 的特点就在于增加了 MINIX 中没有体现的有关操作系统的基本原理, 同时提高了它的通用性与实用性. 因此, 在操作系统教学模型中, MINIX+1.0 已经成为一个比较完整而先进的系统.

参加 MINIX+1.0 系统研制工作的有: 黄祥喜、李小林、郑则仲、赖毅强、梅坚、黎伟强、黄东斌、陈德坚等.

### 参 考 文 献

- 1 Madnick S E, Donovan J J. 石伍锁等译. 操作系统. 北京: 科学出版社. 1980
- 2 Comer D, Fossum T V. Operating System Design, Vol. 1, The Xinu Approach, Prentice Hall, 1988
- 3 Tanenbaum A S. 施振川等译. 操作系统教程. 北京: 世界图书出版公司. 1989

## Design and Implementation of MINIX+1.0, an Experiment Environment of Operating System

*Huang Xiangxi \**

**Abstract** MINIX+1.0 is a modified and augmented version of MINIX, a timesharing, multitask, multiuser operating system developed by A. S. Tanenbaum in 1986. As compared with MINIX, MINIX+1.0 has many new functions, such as process swapping, spooling, management of extended memory, driver of 3.5" floppy disk, et al. MINIX+1.0 can run on IBM PC series computer. As regards to user interface, it is the same as UNIX V7 operating system; both of them have basically the same system calls and shell commands. But their internal structures are very different. MINIX+1.0 has hierarchical architecture, its components communicate by message-passing mechanism. its construction is clear and easy to understand and modify and extend. And, it provides source code of the whole system and shell utilities. Therefore, MINIX+1.0 can provide a good model experiment environment for graduates to study operating system course systematically. On April, 1992, MINIX+1.0 passed authentication of an expert group. The design and implementation of MINIX+1.0 are described in detail in this paper.

**Keywords** operating system, experimental instruction, computer education, MINIX+1.0, MINIX

---

\* Department of Computer Science, Zhongshan University, Guangzhou 510275